What is claimed is:

1. A method of compressing video data having at least one frame having at least one block and each block having an array of pixels, comprising at least one of the steps of:

I) transforming the pixels of each block into coefficients and creating an optimal transmission order of the coefficients;

II) optimizing the speed of processing compressed video data by partitioning the data bitstream and coding each partition independently;

III) predicting fractional pixel motion by selecting an interpolation method for each given plurality of pixels depending upon at least one metric related to each given block; and

IV) enhancing error recovery for a current frame using a frame prior to the frame immediately before the current frame as the only reference frame for lessening quality loss during data transmission.

2. A method of compressing video data according to Claim 1, step I further comprising the steps of:

a) transforming the pixels of each block into coefficients, each coefficient having a coefficient position and a value;

b) determining a position value related to each coefficient position;

c) creating an optimal transmission order of coefficients based on the position values of each coefficient position determined in said step b); and

d) transmitting the coefficients in the order determined in said step c).

3. A method of compressing video data according to Claim 2, said step I further comprising the step of e) dynamically re-ordering the tranmission order of coefficients of said step c) for each frame of video data.

4. A method of compressing video data according to Claim 2, wherein said transforming step a) transforms the pixels into discrete cosine transform coefficients.

5. A method of compressing video data according to Claim 2, said step I further comprising the step of transmitting the tranmission order of coefficients determined in said step c) along with the coefficients transmitted in said step d).

6. A method of compressing video data according to Claim 2, wherein each block has the same number of coefficients and coefficient positions.

7. A method of compressing video data according to Claim 2, wherein each corresponding respective coefficient position conveys the same information from block to block.

8. A method of compressing video data according to Claim 5, said step I further comprising the step of limiting the transmission of coefficient order data to changes in the coefficient order from one frame to the next frame.

9. A method of compressing video data according to Claim 2, said step I further comprising the steps of:

i) consolidating the tranmission order of coefficients of step c) into bands of coefficients, each band having a plurality of coefficients organized by rank in numbers determined in step b); and

ii) transmitting only band information along with the coefficients transmitted in step d).

10. A method of compressing video data according to Claim 9, wherein said step ii) further comprises the step of only transmitting band information when a coefficient changes bands from one frame to the next.

11. A method of compressing video data according to Claim 9, wherein said step ii) further comprises the step of always transmitting all band information.

12. A method of compressing video data according to Claim 2, said step I further comprising the step of providing a key frame which is always completely self-encoded and requires no information from or about a previous frame.

13. A method of compressing video data according to Claim 12, said step I further comprising the steps of:

determining if a given frame is a key frame;

if it is determined that the given frame is a key frame, transmitting the entire tranmission order of coefficients for the key frame; and

if it is determined that the given frame is not a key frame, transmitting only changes in the tranmission order of coefficients from the previous frame to the given frame.

14. A method of compressing video data according to Claim 1, step II further comprising the steps of:

a) dividing the video data into at least two data partitions;

b) selecting an optimal entropy coding method for each data partition;

c) applying the entropy coding methods selected in step b) respectively to each data partition.

15. A method of compressing video data according to Claim 14, wherein step a) further comprises the step of dividing the video data into a predictor token data partition and an error token data partition.

16. A method of compressing video data according to Claim 14, wherein each data partition has selected in step b) a different entropy coding method.

17. A method of compressing video data according to Claim 14, wherein the entropy coding methods selected from in step b) include Huffman coding and arithmetic coding.

18. A method of compressing video data according to Claim 14, further comprising the step of decoding the data partitions asynchronously.

19. A method of compressing video data according to Claim 18, further comprising the step of providing at least two subprocessors, wherein one data partition is decoded by one subprocessor and another data partition is decoded by another subprocessor.

20. A method of compressing video data according to Claim 14, wherein said selecting step b) is performed based on the size of the given data partition.

21. A method of compressing video data according to Claim 15, said step II further comprising the steps of:

i) reading the predictor token data partition;

ii) converting the predictor token data partition into a predictor block;

iii) reading the error token data partition;

iv) converting the error token data partition into coefficients;

v) converting the coefficients into a error block; and

vi) adding the predictor block and the error block to form an image block.

22. A method of compressing video data according to Claim 21, said step II further comprising the step of providing at least two subprocessors, wherein at least one of steps i) -

vi) is performed on one subprocessor and the rest of steps i) - vi) are performed on another subprocessor.

23. A method of compressing video data according to Claim 22, wherein steps iii) and iv) are performed by a fast entropy optimized subprocessor and steps i), ii), v), and vi) are performed by a general purpose subprocessor.

24. A method for optimizing decoder performance of a bitstream produced according to the method of Claim 14 that avoids data and code cache misses, the optimization method comprising the steps of:

i) storing as many distinct functions of the decoder's code as can fit into the code cache;

ii) running the code from step i) for as many blocks as can fit into the data cache;

iii) collecting the next set of distinct functions of the decoder's code;

iv) repeating steps i) - iii) until all of the bitstream has been read and each of the blocks of data have been produced.

25. A method for optimizing decoder performance of a bitstream produced according to Claim 14 that optimizes utilization of subprocessors by assigning each subtask to a separate processor, the method comprising the steps of:

i) running the portion of the decoder that reads error tokens from the bitstream and translates them into coefficients on a fast entropy optimized subprocessor;

ii) running the portion of the decoder that reads the predictor tokens from the bitstream and builds a filtered predictor block from these tokens on a subprocessor with fast access to memory;

iii) running the portion of the decoder that translates the transform coefficients from step i) into an error signal on a subprocessor that has an optimized implementation of the transform coder; and

iv) running the portion of the decoder that adds the predictor block from step ii) to the error signal from step iii) on a subprocessor optimized for motion compensation.

26. A method according to Claim 14, wherein said dividing step a) further comprises the step of dividing the video data into two data partitions, a first data partition representing a first area of the frame and a second data partition representing a second area of the frame.

27. A method according to Claim 26, wherein the first data partition represents an upper half of the frame and the second data partition represents a lower half of the frame.

28. A method according to Claim 26, wherein the first data partition represents a left half of the frame and the second data partition represents a right half of the frame.

29. A method according to Claim 14, wherein said dividing step a) further comprises the step of dividing the video data into three data partitions, each respectively representing level, saturation, and hue information of the frame.

30. A method according to Claim 14, wherein said dividing step a) further comprises the step of dividing the video data into three data partitions, each respectively representing cyan, magenta, and yellow information of the frame.

31. A method of compressing video data according to Claim 1, step III further comprising the steps of:

a) determining the value of the at least one metric associated with a given plurality of pixels to encode;

b) selecting an interpolation method of encoding the given plurality of pixels depending upon the value of the at least one metric determined in step a);

c) applying the interpolation method selected in step b) to the given plurality of pixels to encode; and

d) repeating steps a) - c) for each successive plurality of pixels.

32. A method of compressing video data according to Claim 31, wherein the at least one metric is at least one of motion vector length and a complexity factor.

33. A method of compressing video data according to Claim 31, wherein said step b) selects an interpolation method from the group consisting of bilinear, bicubic, quadratic, and B-spline interpolation.

34. A method of compressing video data according to Claim 31, wherein the given plurality of pixels is an entire frame.

35. A method of compressing video data according to Claim 31, wherein step a) further comprises the steps of:

i) determining if a motion vector length associated with the given plurality of pixels is less than a predetermined length value; and

ii) determining if a complexity factor associated with the given plurality of pixels is greater than a predetermined complexity value.

36. A method of compressing video data according to Claim 35, wherein if the motion vector length associated with the given plurality of pixels is less than the predetermined length value and the complexity factor associated with the given plurality of pixels is greater than the predetermined complexity value, then the interpolation method selected in step b) is bicubic interpolation.

37. A method of compressing video data according to Claim 35, further comprising the step of setting the predetermined length value and the predetermined complexity value one time for a given number of pluralities of pixels.

38. A method of compressing video data according to Claim 37, wherein the setting step is performed once per frame.

39. A method of compressing video data according to Claim 35, wherein the complexity factor of step ii) is a variance of the given plurality of pixels.

40. A method of predicting motion in video data according to Claim 39, wherein the variance is calculated according to the following equation:

$$C = (n \Sigma x_i^2 - (\Sigma x_i)^2)/n^2$$

41. A method of compressing video data according to Claim 1, said step IV further comprising the steps of:

a) using a frame coded prior to the last frame as the only reference frame for a given frame in order to lessen the quality loss associated with transmission over lines which produce lost or corrupt packets; and

b) limiting the application of step a) at least one of periodically and arbitrarily .

42. A method of compressing video data according to Claim 41, further comprising the step of applying steps a) and b) to a video conference.

43. A method of compressing video data according to Claim 42, further comprising the steps of

c) having each party to a video conference compress frames of video data;

d) having each party to the video conference transmit the compressed video data to the other parties with packets that are marked such that the loss or corruption of a packet is detectable; and

e) if any party detects that a packet is lost or corrupted, having the detecting party signal the sending party to send an update frame that has been encoded using a reference frame that has already been successfully received and decoded by all of the remaining parties.

44. A method according to Claim 41, further comprising the steps of:

i) selecting a fixed interval F of video frames;

ii) transmitting this fixed interval F to the decoder;

iii) encoding every F'th frame using only the previous encoded F'th frame for reference;

iv) encoding every non F'th frame using the prior frame as reference; and

v) transmitting each frame of video to the decoder so that loss and corruption are detectable.

45. A method according to Claim 44, wherein said steps i) - v) occur at the encoder.

46. A method according to Claim 45, further comprising the steps of:

vi) receiving coded video data from the encoder;

vii) decoding the video at the decoder; and

viii) if a packet is lost and the lost packet is associated with a non F'th frame, waiting for the next F'th frame to recover the lost packet.

47. A method of compressing video data according to Claim 1, said step IV further comprising the steps of:

encoding a current frame at least one of periodically and arbitrarily at a higher than ambient quality determined by a metric of statistics taken from this and prior coded frames; and

storing the encoded current frame for usage by subsequent frames as a secondary reference frame.

48. A method of compressing video data according to Claim 1, wherein the blocks of a given frame are coded as differences from a similarly sized block in a prior coded frame, comprising the following steps:

a) finding the block that best matches the block to be encoded in a some prior coded frame;

b) determining which of the surrounding 8 blocks provides the closest match to the block we are trying to encode when combined with the block selected in step a); and

c) using a metric on the block selected in step a), the prior coded frame, and a motion vector that distance between the block from a and the block being coded to determine a coefficient.

49. A method of compressing video data according to Claim 1, wherein the blocks of a given frame are coded as differences from a similarly sized block in a prior coded frame, comprising the following steps:

a) finding the block that best matches the block to be encoded in a some prior coded frame;

b) determining the best fractional pixel step away from that best block.

c) calculating a motion vector made from the difference in the row and column between the source block and its best matching block.

d) Uses an algorithm to determining when to encode:

+ no motion vector at all

+ the motion vector by reference to a nearby motion vector

+ the motion vector directly

+ the motion vector as a difference vector from a nearby motion vector

e) Transmitting the motion vector or the difference vector

50. A method as in claim 49 that further comprises the steps of differentially encoding motion vector from the motion vector of the block to the left if that block has a motion vector or the motion vector of the block above if that block has one but the motion vector does not and otherwise encodes the motion vector directly.

51. A method as in claim 49 that further comprises the steps of differentially encoding from a compound motion vector that is calculated by combining motion vector of the block to the left and the block above through an average or a weighted average.

52. A method as in claim 49 that codes the row as differential from the motion vector of the block to the left, and the column as differential from the block above.

53. A method as in claim 51 that only codes the motion vector differentially between the above block or the left block if the motion vectors of the blocks to the left and the blocks above are determined to be similar.